5.7 A 9GHz 65nm Intel Pentium® 4 Processor Integer Execution Core

Sapumal Wijeratne, Nanda Siddaiah, Sanu Mathew, Mark Anders, Ram Krishnamurthy, Jeremy Anderson, Seung Hwang, Matthew Ernest, Mark Nardin

Intel, Hillsboro, OR

The 64b integer core of the Intel Pentium® 4 processor shown in Fig. 5.7.1 uses a 2× frequency clock to enable single-cycle latency on the critical ALU bypass loops [1]. Domino circuitry optimized for 2× frequency operation replaces low voltage swing (LVS) circuit technology [2], because process-scaling studies indicated that transitioning LVS to a 65nm CMOS process [3] will only yield a speedup of 14%–an amount that is insufficient to enable the frequency targets of a 4th-generation Intel Pentium® 4. The circuit architectures implemented in the ALUs, AGU, and integer register file enable a 9GHz/1.3V frequency target while reducing power consumption, relative to a 90nm version of this product described in [4].

Figure 5.7.2 shows the domino chain topology and single-phase pulsed clock employed to generate the 2× frequency clock. Limiting each phase to 2 domino gates terminated with an SDL sequential minimizes the risk of pulse evaporation. Since the risk of clock-pulse evaporation prohibits the use of clock wave shaping, up to one inversion of power contention is allowed at the start and end of domino precharge. Special post-processing tools enforce bandwidth compliance by ensuring >90% voltage swing on all domino and clock nodes. These tools comprehend loss in pulse width due to global clock duty-cycle deviations, power races, time borrowing, and statistical variation of the pulsed clock. At high-frequency operation, domino-gate sizing is dictated primarily by these bandwidth/edge-rate constraints rather than by forward-delay requirements. The use of extra-high-performance transistors is avoided to minimize leakage power.

The 2× frequency dual-rail ALU/AGU front-end (Fig. 5.7.3), selects and conditions (set/zero/invert) each operand from 5 sources (Fig. 5.7.1) using a 5:1 merged multiplexer-latch. This mux-latch is segmented with a default write zero function added to each segment and the segments are merged with a distributed domino NOR. This configuration simultaneously minimizes both multiplexer-transfer-gate size and delay, yielding a front-end design that is faster than the earlier LVS implementation by one gate delay, with no area penalty. This speed advantage compensates for poor RC scaling of the source busses and also enables the replacement of the previously dual-rail L0 cache with a dense single-rail design, enabling the 65nm domino integer core design to fit within the footprint of the scaled 90nm LVS version. A pseudo-asynchronous mux-latch clock-gating scheme enables single-cycle granularity clock gating in the ALU/AGU cycle-1 datapath, resulting in lower dynamic power. Elimination of LVS reset and race constraints enable this level of clock gating to be carried into the subsequent cycles of AGU/ALU datapath.

The 32b adder core of each ALU employs a sparse-tree circuit that generates every fourth carry using a radix-2 carry-merge tree with 4b conditional sums generated using ripple-carry adders [5]. The ALU requires 2 cycles to execute an ADD/SUB instruction using a fully time borrowing 16-stage dual-rail domino design. The reduced propagate and generate (PG) fanouts of the sparse-tree, along with clock gating, enable 51% reduction in normalized dynamic power over the previous LVS adder. The domino adder area is 5% larger than a scaled 90nm LVS adder.

The address generation unit (AGU) computes the linear address for cache access from 4 source operands. These operands are first merged into a single carry-save result using a 4:2 compressor (Fig. 5.7.4a), followed by a sparse-tree completion adder that produces the final linear address. The lower 16b of the sparse-tree (Fig. 5.7.4b) includes a PG stage, followed by 4 stages of carry-merge (CM). The resulting 1-in-4 carries select the correct conditional sums. To support 16b addressing, a lower 16b carry-out is needed from the sum of only the base, index, and displacement. By using an extra sparse carry-tree to subtract the segment base from the linear address, the extra carry-out is obtained 3 gate delays after the linear address. This technique minimally affects the critical early-arriving lower 16b address, needed for the cache tag, by loading only the non-critical 4b conditional sum circuits. The sparse-tree adder organization allows the subtract operation to begin in the conditional sum blocks (Fig. 5.7.4c) before the linear address computation is complete. Within each 4b conditional sum circuit, 2 additional ripple-carry chains, one with carry-in of 1 and the other with carry-in of 0, create conditional PG signals, respectively, for the subtraction. An additional MUX chooses the correct PG signals, which are then merged in 2 stages of CM shown in Fig. 5.7.4b to yield the MUX select for the upper 16b linear address. Compared to the LVS AGU design, the sparse-tree design, ripple-carry subtract, and efficient clock-gating enable 59% normalized dynamic power reduction and 10% area reduction.

A conventional rotator architecture using a single, wide MUX cannot perform single-cycle rotate/shift operations and meet bandwidth requirements at 2× frequency. As a result, the rotate/shift function is broken into 3 domino MUX stages (Fig. 5.7.5). The first MUX stage rotates each byte 0-7 positions, and the second MUX rotates 0, 8, 16, or 24 positions. The final MUX completes shift operations with a zero or size appropriate MSB fill, or passes the input through if the op is a rotate. Low-latency operation is enabled by performing a speculative rotate in the first MUX, followed by a correction in the second MUX. The byte-wise rotate of the first MUX results in some bits out of position by a byte. Control logic for the second MUX detects out-of-position bits and modifies rotate amounts to compensate. This architecture provides decreased gate load, wire length, and wire count for critical data/control signals at a 30% area penalty over the scaled LVS design.

The 144-entry 6r3w register file (Fig. 5.7.6) uses the 2× frequency clock to obtain a throughput of 12r6w per processor clock. Two-cycle write operation is achieved by writing half of the array each cycle; however, read operations are optimized to complete in one 2× frequency clock cycle. A self-terminating precharge method enables forward time borrowing in order to maximize evaluate bandwidth of the high RC GBL and initiate the GBL read as soon as possible.

Frequency shmoo (Fig. 5.7.7) obtained using a debug feature that allows isolated testing of the integer core shows that the 65nm Intel Pentium® 4 integer core frequency exceeds 9GHz at 1.30V 70°C ambient on typical initial revision Si. At this frequency/V the integer core power consumption is 10.36W. AGU/ALU normalized switching capacitance and normalized leakage power are reduced 50% and 72%, respectively, over previous generation LVS technology. In addition to enabling faster frequencies, especially in dual-core variants of this microprocessor, this redesign of the integer core, that is at the center of the processor hot spot, is estimated to reduce the maximum microprocessor temperature by 8°C.

Acknowledgments:
The authors thank J. Schutz, S. Borkar, M. Haycock, J. Rattner, S. Pawlowski, D. Guddat, G. Taylor, C. Webb, D. Deleganes and several team members from Logic Technology Development and Circuit Research Labs at Intel for support and contributions.

References:
[1] D. Sager, et al., "A 0.18μm CMOS IA32 Microprocessor with a 4GHz Integer Execution Unit," *ISSCC Dig. Tech. Papers*, pp. 324-325, Feb., 2001.
[2] D. Deleganes, et al., "Low-Voltage-Swing Logic Circuits for a 7GHz x86 Integer Core," *ISSCC Dig. Tech. Papers*, pp. 154-155, Feb., 2004.
[3] P. Bai, et al., "A 65nm Logic Technology Featuring 35nm Gate Lengths, Enhanced Channel Strain, 8 Cu Interconnect Layers, Low-k ILD and 0.57μm² SRAM Cell," *IEDM Technical Digest*, pp. 657-660, Dec., 2004.
[4] J. Schutz, et al., "A Scalable X86 CPU Design for 90 nm Process," *ISSCC Dig. of Tech. Papers*, pp. 62-63, Feb., 2004.
[5] S. Mathew, et al., "A 4-GHz 130-nm Address Generation Unit with 32-bit Sparse-Tree Adder Core," *IEEE J. Solid-State Circuits*, vol. 38, pp. 689-695, May, 2003.

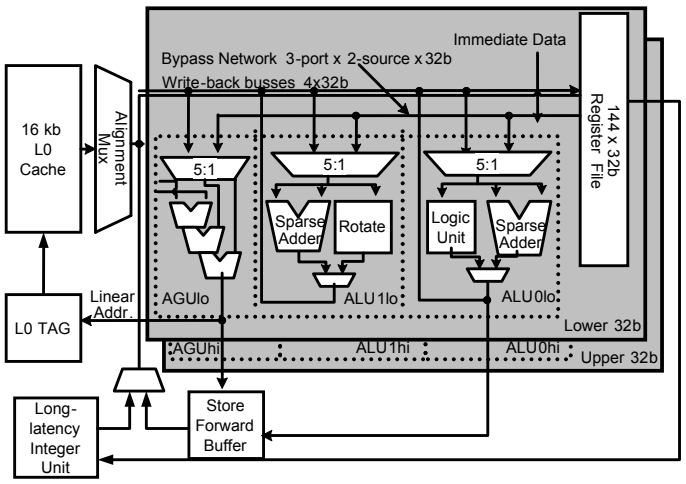

Figure 5.7.1: Integer execution core organization.

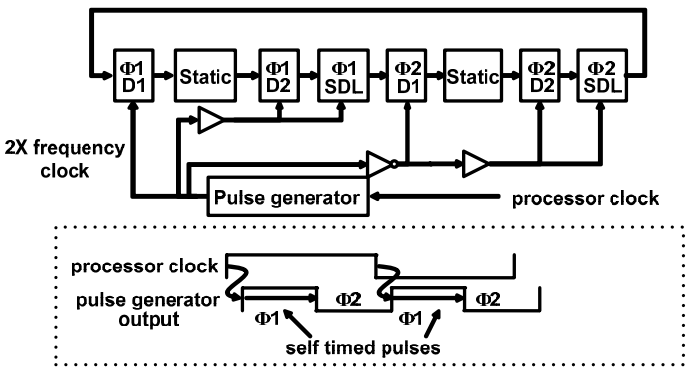

Figure 5.7.2: 2× frequency domino and clock scheme.

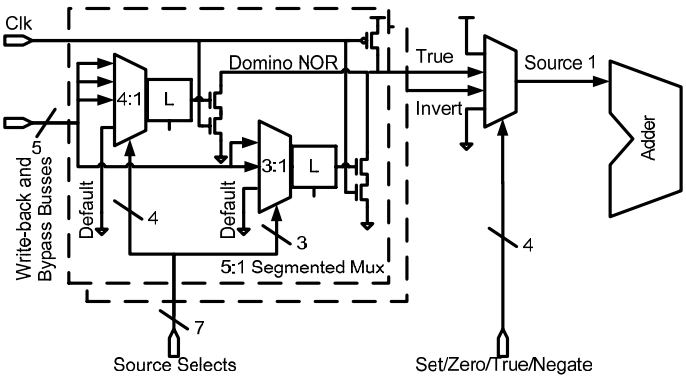

Figure 5.7.3: AGU/ALU front-end design.

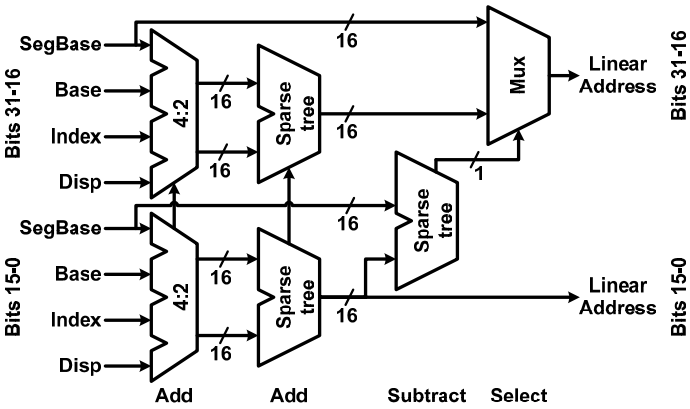

Figure 5.7.4a: AGU organization.

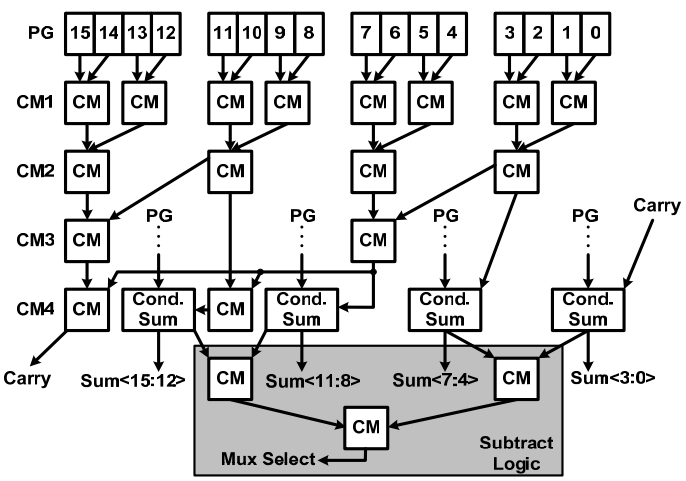

Figure 5.7.4b: Critical AGU 16b sparse carry-merge tree.

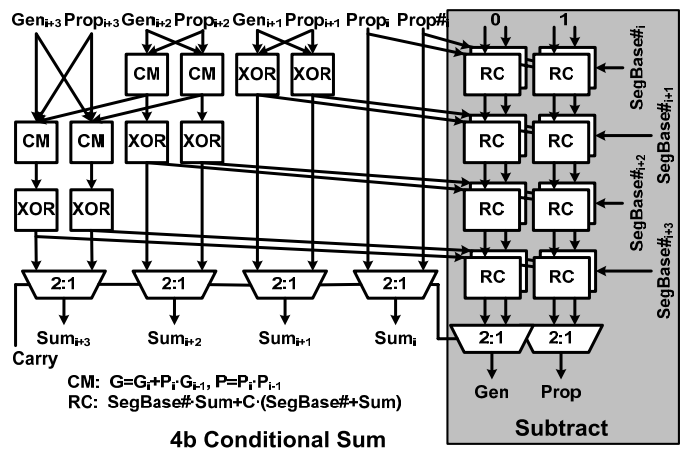

Figure 5.7.4c: AGU 4b conditional sum-generators.

Continued on Page 642

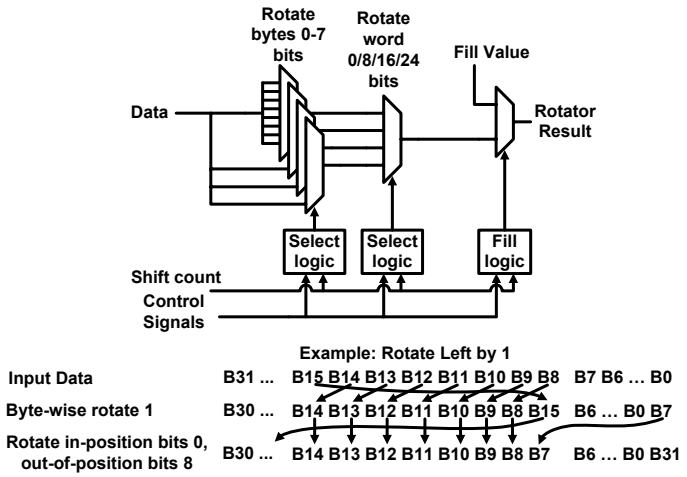

Figure 5.7.5: Rotator design.

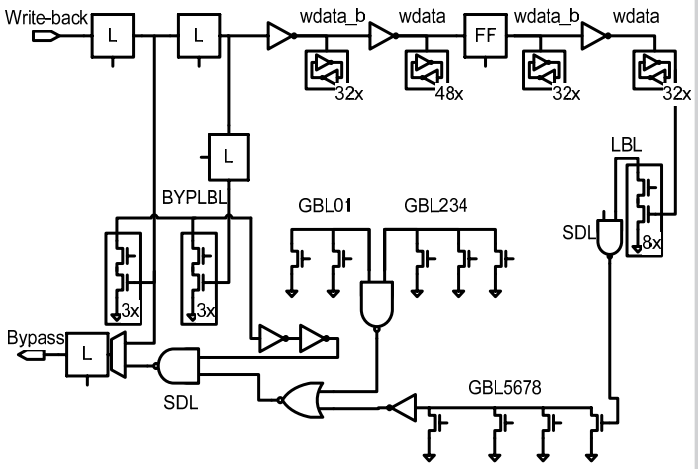

Figure 5.7.6: Integer register file.

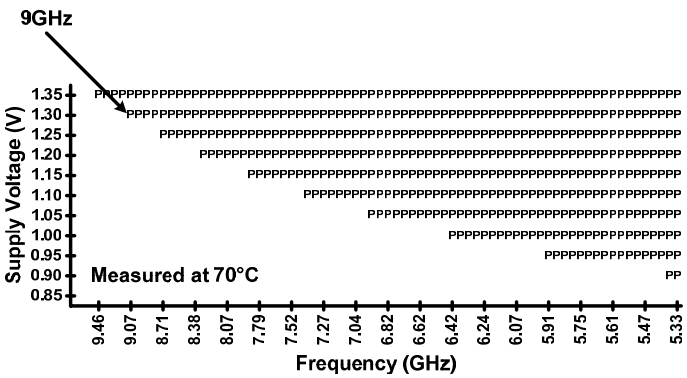

Figure 5.7.7: Shmoo / measurement results.

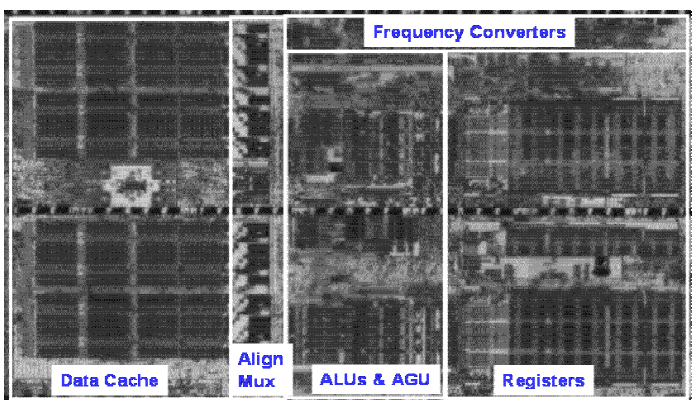

Figure 5.7.8: Die micrograph showing outlines of integer core blocks.

